SEMANTIC RECONSTRUCTION: A COLMAP - UNET FUSION WITH VOTING

Srijan Pal, Amitabha Deb, Roozbeh Ehsani & Tejasvi Bansal
CSCI 5561:Computer Vision Project Final Report
University of Minnesota, Twin Cities

1 INTRODUCTION

Semantic reconstruction helps machines to better understand and interact with the world. It has various practical applications like autonomous navigation and perception for self-driving vehicles, robot perception, urban planning, indoor floor mapping, and medical image analysis. While 2D semantic segmentation has achieved high accuracy comparable to human-level performance, 3D semantic reconstruction has a lot of challenges due to the complexity of scenes, quality, and quantity of input data.

In this project, we conducted a 3D reconstruction of an environment while preserving the semantic information. We constructed a pipeline from 2D semantic labels and 3D reconstructed points to estimate 3D semantic reconstruction.

2 LITERATURE REVIEW

Image segmentation is a vital computer vision technique that enables the precise identification and delineation of objects of regions of interest within digital images. Mask R-CNNHe et al. (2017) is a state-of-the-art deep learning architecture used for instance segmentation, which is the task of not only detecting objects in an image but also segmenting them into precise pixel-level masks. It builds upon the Faster R-CNNRen et al. (2015) architecture, which is a widely used framework for object detection. At a high level, Mask R-CNN consists of several modules. Backbone is a standard convolution neural network (typically, ResNet50 or ResNet101)He et al. (2015) that serves as a feature extractor. Feature Pyramid Network (FPN) improves the standard feature extraction pyramid by adding a second pyramid that takes the high-level features from the first pyramid and passes them down to the lower layer. Our implementation of Mask R-CNN uses a ResNet101 + FPN backbone. The Region Proposal Network (RPN) is a lightweight neural network that scans the image in a sliding window fashion and finds areas that contain objects. The regions that the RPN scans over are called anchors. There are about 2,000 anchors of different sizes and aspect ratios, and they overlap to cover as much of the image as possible. The RPN generates two outputs for each anchor: class of anchor that can be foreground (FG) or background (BG), and bounding box. The next stage runs on the regions of interest (ROIs) proposed by the RPN, and just like the RPN, it generates two outputs for each ROI: the class of the object in the ROI which can belong to specific classes (person, car, chair, ...), and Bounding Box Refinement which further refine the location and the size of it to contain the object. ROIs that are BG will be discarded. ROI pooling refers to cropping a part of a feature map and resizing it to a fixed size. The segmentation mask branch is a convolutional network that takes the positive regions selected by the ROI classifier and generates masks for them. The resolution of masks is low but they are soft masks, represented by float numbers. The small mask size helps keep the mask branch light.

The U-Net paper Ronneberger et al. (2015) introduces a convolutional neural network architecture designed for biomedical image segmentation tasks. Notable for its U-shaped architecture, the model combines contracting and expansive paths to capture contextual information and refine segmentation. Skip connections aid in preserving spatial information during the upsampling process. U-Net has proven effective in various medical image segmentation applications, providing accurate delineation of structures in biomedical images. The architecture's versatility and success in segmentation tasks have made it a pivotal contribution to the field of deep learning in medical imaging.

In the paper 'Segment Anything'Kirillov et al. (2023), a new task, model, and dataset for image segmentation has been introduced. Segment Anything Model (SAM) has learned a general notion of what objects are – this understanding enables zero-shot generalization to unfamiliar objects and images without requiring additional training.

As described by Schronberger and Frahm in the work "Structure-from-Motion Revisited"Szeliski (2011), COLMAP is an image-based 3D modeling Pipeline that takes sets of images of a particular scene and their corresponding camera parameters as input and generates a 3D reconstructed map of the scene.

A naive approach for performing semantic reconstruction is to apply a majority voting or frequency-based method to determine the most frequently occurring label for every voxel/point from the corresponding pixels in the 2D images. With the rise of deep learning models and the increase in the volume of data, many networks were also created for this purpose.

3D-UNet Çiçek et al. (2016) is used for volumetric segmentation using a few annotated 2D slices. It can generate 3D volume segmentation semi-automatically with the 3D voxels and a few annotated slices as input and fully-automatically by training the model on representative data and then providing the voxel as input. The network generalizes easily, but data preparation & training is required for it to be successful. PointNetCharles et al. (2017) uses point clouds as input and gives class labels for each point. It utilizes symmetric operations such as max pooling and data-dependent spatial networks to get a lower dimensional skeleton of the point cloud. The network is highly robust to noise and fast due to the permutation invariance of points in the input. Pointnet++Qi et al. (2017) further improves the performance by using a hierarchical network that uses PointNet recursively. In the paper "Learning to Segment 3D Point Clouds in 2D Image Space"Lyu et al. (2020) the authors focus on effectively projecting 3D data to 2D using graphs but setting the points as nodes and the distance as edges. The 2D images are then segmented using U-Net.

3 Approach

Our methodology involves the generation of 3D sparse and dense point clouds using COLMAP, extraction of the segmentation masks using U-Net from the images provided, and finally employing a voting mechanism paired with a k-means algorithm to perform 3d semantic reconstruction.

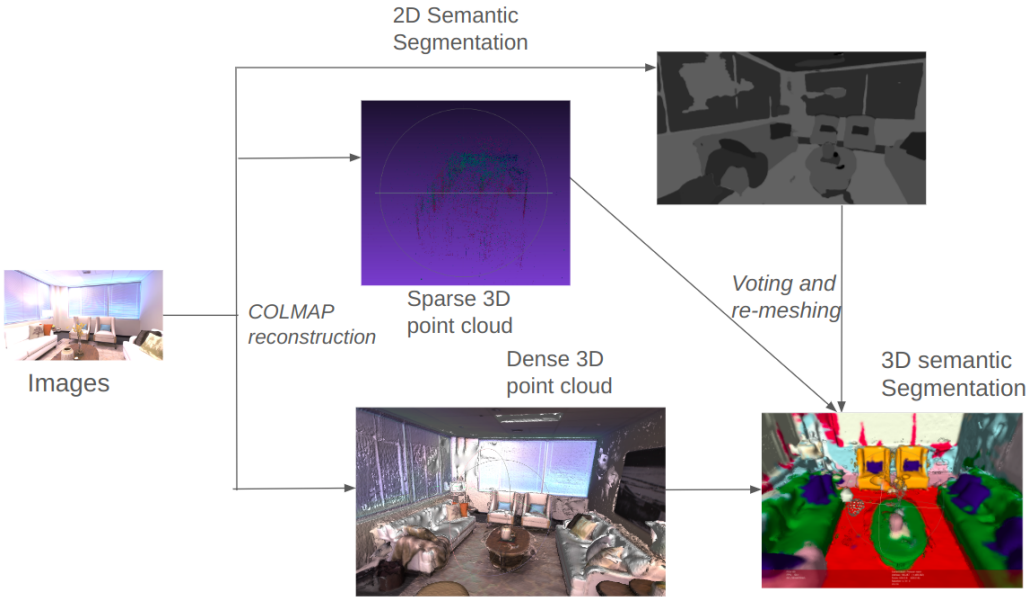

Figure 1: Semantic Reconstruction Methodology

3.1 3D POINT CLOUD

For the 3D point cloud generation and 3D reconstruction, we are primarily using the open-source application COLMAP.

COLMAP uses feature extraction algorithms to identify key points in an image and feature matching algorithms to find correspondences between images. These correspondences are used to generate a sparse reconstruction of the scene.

Sparse reconstruction uses a portion of all the available 2D points to generate an initial estimate which helps in building the foundation for the next phase, the dense reconstruction. The sparse reconstruction helps in optimizing the camera positions in the 3D coordinate.

Subsequently, the dense reconstruction phase takes the result from the sparse modeling stage to recover a dense model of the scene in the form of dense 3D point clouds. It uses a multi-view stereo process for estimating the depth and creating a dense point cloud. This can be further used to generate surface reconstruction using 3D surface mesh. We can have multiple types of output from COLMAP like dense depth maps, dense 3D point clouds, and 3D mesh.

We also tried doing the 3D reconstruction directly on semantically segmented images. However, the main problem will be that the semantically labeled images will lose their distinctive features as each class is marked by solid colors. The first step of COLMAP which involves feature extraction and feature matching between the images will not work properly.

3.2 SEGMENTATION (U-NET)

In order to do segmentation of 2D images, the U-net algorithm has been employed Ronneberger et al. (2015). It is characterized by its symmetric, U-shaped architecture, which includes a contracting path (encoder) to capture context and a symmetric expanding path (decoder) that enables precise localization 2. The contracting path uses a standard design similar to that of a typical convolutional network. It consists of the repeated application of two 3×3 convolutions, each followed by a rectified linear unit (ReLU) and a 2×2 max pooling operation with stride 2 for downsampling. At each downsampling step, we double the number of feature channels (i.e., the process begins with two convolution layers of 16. After each downsampling step, the number of layers doubles: first to 32, then 64, 128, and finally 256). Every step in the expansive path includes an upsampling of the feature map followed by a 2×2 transposed convolution that halves the number of feature channels (i.e., the process onset with 256, then gets halved to 128, 64, 32, and at the end 16), a concatenation with the correspondingly cropped feature map from the contracting path, and two 3×3 convolution, each followed by a ReLU. The network's final layer employs a 1x1 convolution to transform each 16-component feature vector into the specified number of classes. Overall, there are 23 convolutional layers in the network. The input image fed into the algorithm is a grayscale image with dimensions $688\times1200\times1$. The resulting segmentation map produced by the algorithm has dimensions of $688\times1200\times29$, where each of the 29 channels corresponds to the detection of an object in the input image, utilizing a one-hot encoding approach.

3.3 SEMANTIC RECONSTRUCTION

For the semantic enrichment of the 3D point cloud, we employ the voting mechanism and the k-nearest neighbourMucherino et al. (2009) classification algorithm. The voting mechanism is a technique employed in 3D semantic reconstruction to aggregate labels from multiple sources or viewpoints, aiming to enhance the accuracy and completeness of the reconstructed 3D scene. This technique involves a voting process where each source contributes its observations, and the final label is determined based on the accumulated votes.

Our methodology begins with the judicious application of the voting mechanism to assign semantic labels to sparse 3D points extracted from the points3d.txt file. These sparse points represent key features in the environment, specified by their X, Y, and Z coordinates, along with an essential identifier linking them to the source image. Leveraging information from the images.txt file, we seamlessly correlate these sparse points to their originating images, extracting the corresponding pixel coordinates (X, Y) on each image.

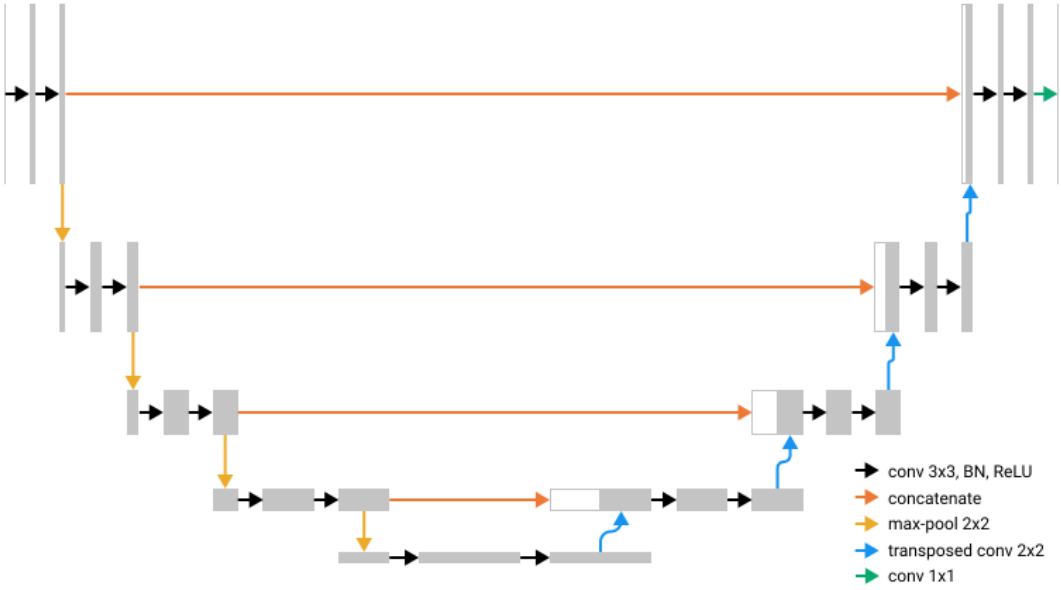

Figure 2: U-net architecture. Each gray box shows a multi-channel feature map, while the white boxes indicate feature maps that have been copied. The arrows illustrate the various operations involved.

With this wealth of pixel-level information at our disposal, the subsequent challenge lies in determining the semantic label for each 3D point. Employing a majority voting strategy, we collectively analyze the semantic labels associated with the corresponding pixels, making informed decisions regarding the semantic classification of the sparse 3D points. This process, by aggregating votes across the image sources, robustly refines the semantic representation of the sparse point cloud, ensuring a holistic understanding of the scene.

Advancing beyond sparse point clouds, our research delves into the classification of dense 3D points, a critical component for achieving a nuanced semantic reconstruction. Here, the kNN classification algorithm takes the lead, considering the nearest k sparse points for each dense 3D point. This localized and contextualized approach ensures that the semantics assigned to dense points are influenced by the immediate surroundings, enhancing the granularity of our semantic reconstruction.

In essence, our approach seamlessly integrates the voting mechanism with k-means classification, crafting a comprehensive framework for semantic reconstructions. By fusing information from sparse and dense points, we not only address the intricacies associated with sparse data but also ensure a sophisticated semantic understanding of the entire scene.

4 Experiments and Results

We are using room_0 for performing the semantic reconstruction.

4.1 COLMAP Results

The baseline approach for this project involves using off-the-shelf products. The first step was to generate the 3D sparse and dense point clouds from the images using COLMAP, which uses both Structures from Motion and Multi-View Stereo. The COLMAP GUI has a feature named 'Automatic Reconstruction'. A window pops up that lets you select the workspace folder, the folder containing the sequential images, and the folder with the corresponding masks. There are options to generate the sparse and the dense point clouds step by step or both the sparse and dense point clouds simultaneously. There is also an option if we want to use GPU for doing the computation.

The automatic reconstruction in COLMAP of room_0 (containing 500 images) took about 4-5 hours to compute both the sparse and dense point cloud at high precision using GPU. The sparse point cloud was then exported as text files (which contain the 'images.txt', 'points3D.txt', and 'camera.txt' files) containing all the information needed for semantic segmentation.

Once the semantic segmentation-based voting is done, the colors of each dense 3D point are changed based on their corresponding class labels (the voting technique is further discussed in detail in section 4.3.2 Dense Point Cloud Labelling). This labeled dense 3D point cloud was further reconstructed to make a semantic reconstructed map of the scene using another software called Meshlab.

After doing the semantic reconstruction, we did some cleanup on the 3D point cloud using cleanup tools in Meshlab, like computing normal for point sets and removing isolated pieces, vertices, faces, etc. This meshed using Poisson Surface reconstruction to get the semantically labeled 3D map of the scene.

The sparse 3D point cloud and the dense 3D point cloud of room_0 generated from COLMAP are provided in Figure 3.

Figure 3: Sparse and Dense Point Cloud generated using COLMAP

4.2 Segmentation Results

The Room-0 dataset was utilized for segmentation tasks. This dataset was divided, allocating 315 images for training the U-net model, while a separate set of 50 images was designated for testing and validation purposes. The training of the model was conducted with a batch size of 5 over 50 epochs. The Adam optimizer was employed due to its efficiency in handling multi-class problems, and categorical cross-entropy was chosen as the loss function, aligning with the model's output of 29 classes.

As depicted in Figure 4 (Left), the model's loss, measured by categorical cross-entropy, demonstrates a decreasing trend across the 50 epochs. This decrease signifies the model's increasing accuracy in predicting the data. Conversely, Figure 4 (Right) illustrates the model's accuracy over the same period, underscoring a consistent improvement in the rate of correct predictions. By the conclusion of the 50 epochs, the model achieved an accuracy of 97%. Additionally, the mean Intersection over Union (IoU) value attained was 75%, further indicating the model's effectiveness in segmentation tasks.

Figure 5 presents a comparative analysis between the annotated ground truth for semantic segmentation and the outcomes produced by the model on the test images. This comparison clearly illustrates the efficacy of the U-net model in accurately performing semantic segmentation on the input images.

4.3 Semantic Reconstruction Results

The 3D sparse and dense point clouds used for semantic reconstruction were obtained through COLMAP, capturing the intricate details of the scene. Additionally, U-Net output masks were employed to refine and enrich semantic information.

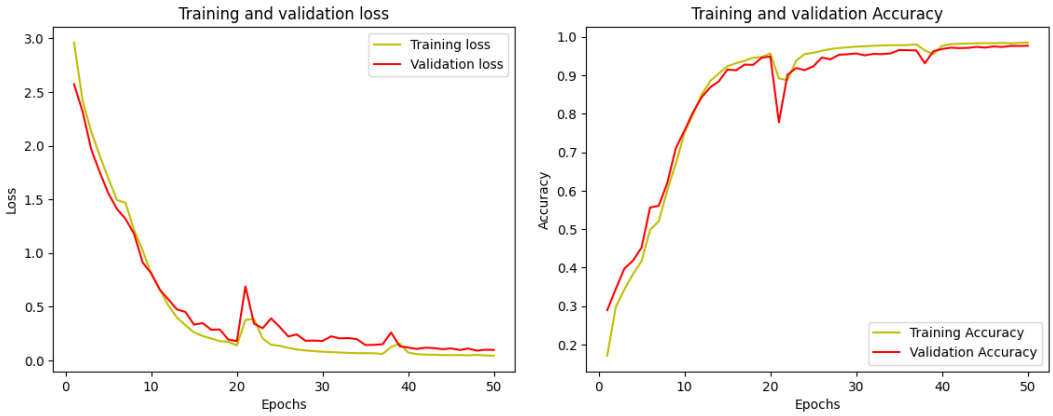

Figure 4: Left figure: Model's loss over 50 epochs. Right figure: Accuracy progression across 50 epochs

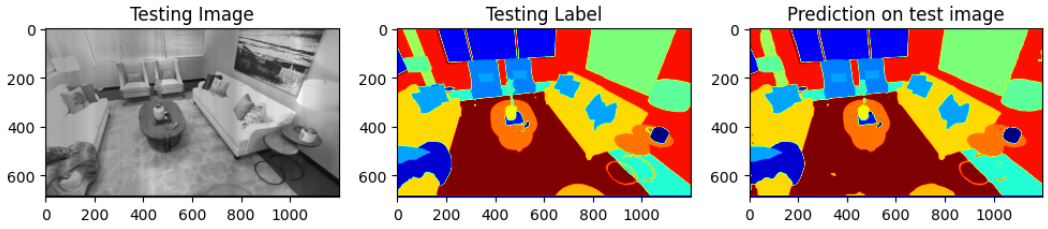

Figure 5: Comparison between ground truth semantic segmentation and model's segmentation output. Left figure: Input image of the testing dataset. Middle figure: Ground truth semantic segmentation. Right figure: Output of the U-net model.

The direct application of a majority voting mechanism to sparse point clouds facilitated label assignment using corresponding masks. For kNN classification, 10 nearest neighbors from the sparse point cloud were considered for each dense point.

4.3.1 Sparse Point Cloud Labeling

A direct application of majority voting was employed using masks generated by U-Net, facilitating the semantic labeling of sparse point clouds.

Visualization of the sparse point cloud labels revealed distinct object boundaries. The success of this method is evident in the formation of clusters comprising similarly colored points with identifiable shapes. This clear delineation highlights the effectiveness of the majority voting approach in capturing semantic information and enhancing object recognition within the sparse point cloud.

This experiment emphasizes the successful utilization of majority voting for semantic labeling, providing clarity and recognition in the visualization of sparse point clouds.

4.3.2 Dense Point Cloud Labeling

The k-means algorithm was applied, considering the 10 nearest sparse neighbors to derive semantic labels for dense points within the point cloud.

Visualization of the results demonstrated the successful recognition of objects within the dense point cloud. The discernible and accurate identification of objects showcased the robustness of the k-means classification approach. Moreover, the use of distinct colors representing various semantic classes further emphasized the algorithm's ability to differentiate and categorize objects effectively.

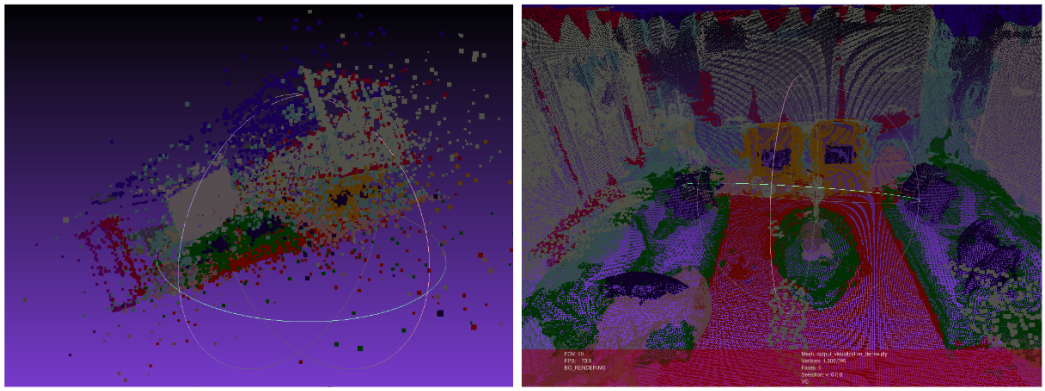

Figure 6: Semantic Reconstruction Sparse and Dense Results

This experiment underscores the effectiveness of the kNN classification in enhancing semantic understanding within dense point clouds. The achieved recognition and distinguish-ability of objects contribute to the overall success of our 3D semantic reconstruction methodology.

The results thus demonstrate that voting can be very effective in this task. But this is limited by the quality of points generated by COLMAP and the quality of the output masks. Further, k-means classification can fail near the edges between two objects, where more sparse points of another object is closer to the dense point. This can result in less accurate boundaries.Deep Learning based semantic reconstruction using the point clouds might help overcome these issues due to the learning approach it employs.

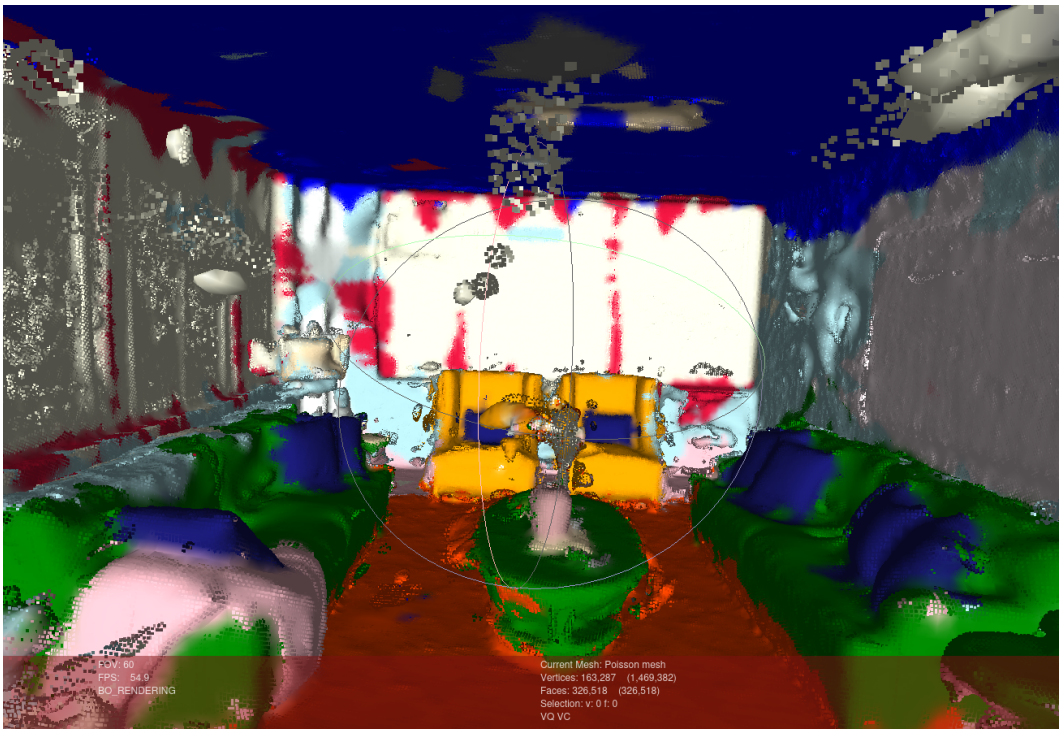

Figure 7: Semantic Reconstruction Result

5 CONCLUSIONS

We have worked on a voted semantic reconstruction, where we successfully obtain the class labels for the point cloud from all the corresponding segmented images. We can see that the method is effective, but it has its limitations. To address these we can explore whether deep learning models such as pointnet++ can be used directly for obtaining 3d semantic labels from the COLMAP output.

In conclusion, 3D semantic reconstruction emerges as a pivotal field, integrating spatial understanding with semantic richness. The fusion of sparse and dense point clouds, coupled with voting mechanisms and k-means classification, facilitates nuanced scene interpretation. Challenges include refining sparse data representation and optimizing computational efficiency. Future endeavors should prioritize addressing these hurdles while advancing semantic reconstruction applications in robotics, urban planning, and immersive technologies, unlocking its full potential for comprehensive spatial understanding.

6 CONTRIBUTIONS

Amitabha - Voting, K-means
Srijan - COLMAP, Voting
Tejas - SAM, Masked RCNN
Roozbeh - My main contribution was to the segmentation of the 2D images. At first, MaskR-CNN was elected as the segmentation algorithm, and pre-trained model based on the coco dataset was used but the accuracy was low. As a result, we decided to train the model based on the room-0 dataset. In order to train it, we needed .JSON file that shows the location of each object (i.e., the coordinate of a polygon that covers the object) in the image. The format of the provided annotated images was not compatible with .JSON file and even after writing a code to convert the provided annotated images into .JSON file it didn't work. Because of time constraints, we could not make .JSON file from scratch(room-0 has 29 different classes). Finally, we came up with the idea of using U-net (provided annotated images can be used).
Throughout this course, I have learned new segmentation and detection architectures including family RCNNs, SAM, and U-net.

Code and Data: Github

ACKNOWLEDGEMENT

We would like extend our gratitud thank our professor Volkan Isler

REFERENCES

R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85, 2017. doi: 10.1109/CVPR.2017.16.

Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: Learning dense volumetric segmentation from sparse annotation. *CoRR*, abs/1606.06650, 2016. URL http://arxiv.org/abs/1606.06650.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. URL http://arxiv.org/abs/1703.06870.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023.

Yecheng Lyu, Xinming Huang, and Ziming Zhang. Learning to segment 3d point clouds in 2d image space. *CoRR*, abs/2003.05593, 2020. URL https://arxiv.org/abs/2003.05593.

Antonio Mucherino, Petraq J. Papajorgji, and Panos M. Pardalos. *k-Nearest Neighbor Classification*, pp. 83–106. Springer New York, New York, NY, 2009. ISBN 978-0-387-88615-2. doi: 10.1007/978-0-387-88615-2_4. URL https://doi.org/10.1007/978-0-387-88615-2_4.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *CoRR*, abs/1706.02413, 2017. URL http://arxiv.org/abs/1706.02413.

Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015. URL http://arxiv.org/abs/1506.01497.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241. Springer, 2015.

Richard Szeliski. *Structure from motion*, pp. 303–334. Springer London, London, 2011. ISBN 978-1-84882-935-0. doi: 10.1007/978-1-84882-935-0_7. URL https://doi.org/10.1007/978-1-84882-935-0_7.